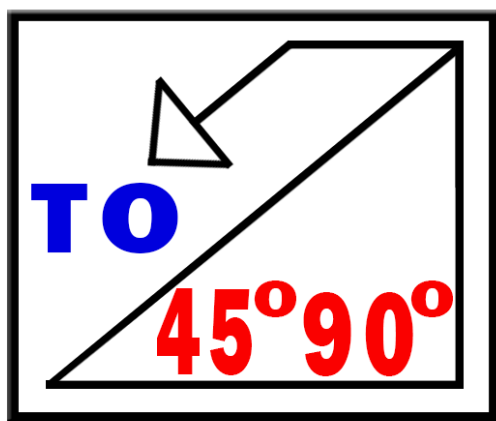


# PCS

## Programando com S.A.S

### PROGRAMAÇÃO FUNCIONAL – LOGO

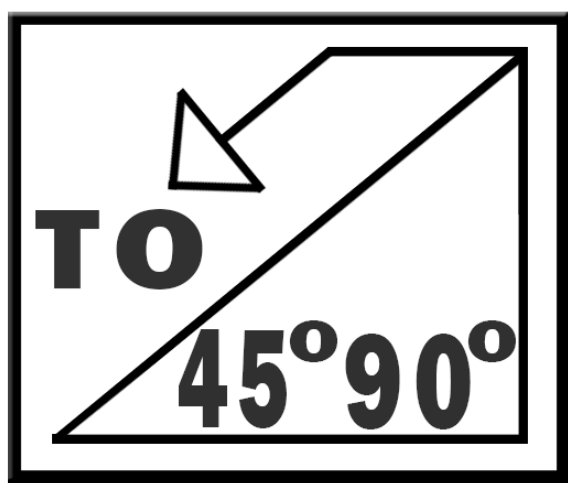


Sualé Abdul Sualé

# **PCS**

## **Programando com S.A.S**

### **PROGRAMAÇÃO FUNCIONAL – LOGO**



**Sualé Abdul Sualé**

## **Ficha Técnica**

<b>Autor:</b>	Sualé Abdul Sualé
<b>Título:</b>	Programação Funcional - LOGO
<b>Editora:</b>	S/E
<b>Local de Publicação:</b>	Maputo
<b>Data de Publicação:</b>	09 de Fevereiro de 2017
<b>Número de Páginas:</b>	29

# Índice

Capítulo I.....	1
Origem da Linguagem.....	1
Conceito.....	1
Classificação das Linguagens de Programação.....	1
Linguagem de Baixo Nível ou Assemblers.....	1
Linguagem de Alto Nível.....	2
Características de uma Linguagem Funcional.....	2
Capítulo II.....	3
Baixando o Compilador.....	3
Instalação dos Compilador.....	7
Instalação de MSWLogo.....	7
Instalação de SuperLogo.....	11
Capítulo III.....	16
Comandos Básicos.....	16
Estrutura de um Programa em Logo.....	16
Visualização do Programa em Logo.....	17
Demonstração de criação e visualização de um Programa em Logo.....	17
Outros comandos.....	18
Capítulo IV.....	20
Exercícios.....	20
Referência Bibliográfica.....	22
Biografia.....	23
Anexos.....	24

# Capítulo I

## Origem da Linguagem

A linguagem **LOGO** conhecida por *tartaruga*, foi criada por Wally Feurzieg e Seymour Papert, em 1967, tendo o paradigma funcional, procedural e reflexão. Esta linguagem foi influenciada por LISP que por sua vez influenciou Smalltalk, Etoys, Scratch e NetLogo, e os compiladores usados são vários, citando alguns (Super Logo, MSWLogo, UCBlogo entre outros).

A linguagem é voltada para crianças, jovens e adultos, esta usada como uma ferramenta de apoio ao ensino regular e os iniciantes em programação de computadores.

O seu ambiente gráfico envolve uma tartaruga ou robô pronto para responder todos os comandos do usuário, sendo esta uma linguagem interativa, assim que for digitado o comando o seu resultado é visualizado imediatamente. Anteriormente, a maioria dos comandos deferiam-se a *desenhar* e *pintar*, mas actualmente pode-se trabalhar com textos, fórmulas e até Inteligência Artificial, sendo esta uma excelente ferramenta para o ensino regular.

## Conceito

A linguagem Logo é uma linguagem de programação funcional.

Programa é um conjunto de instruções a serem seguidas e executadas.

Programa de Computador é um conjunto de dados escritos numa determinada linguagem de programação.

Linguagem de Programação é um conjunto de ferramentas que nos permitem escrever um programa de computador destinado a instruí-lo para realizar uma determinada tarefa.

## Classificação das Linguagens de Programação

As linguagens de programação dividem-se em:

- ✓ Linguagem de Baixo Nível ou Assemblers
- ✓ Linguagem de Alto Nível

### Linguagem de Baixo Nível ou Assemblers

São linguagens baseadas em código máquina, executadas directamente pela CPU obedecendo as instruções que contêm um código de operação e endereços de memórias que podem ser um ou mais, encontrando-se ao nível de Hardware.

## **Linguagem de Alto Nível**

São linguagens que nos permite programar através de comandos com palavras da língua Inglesa.

### **Características de uma Linguagem Funcional**

As linguagens funcionais são caracterizadas por:

- ✓ Não possuir um objectivo delimitado;
- ✓ Possuir uma orientação;
- ✓ Ser uma função;

Sendo assim, é importante descrever que encontramos características como:

- ✓ Passagem de funções por parâmetro;
- ✓ Retorno de funções por outra função;
- ✓ Composição de funções;
- ✓ Funções anónimas; e
- ✓ Chamadas parciais de funções.

## Capítulo II

### Baixando o Compilador

#### 1º Passo

Vai ao seu navegador e pesquise por um motor de busca “é habitual usar *Google*”, e em seguida pesquise por:

- ✓ MSWLogo; ou
- ✓ SuperLogo.

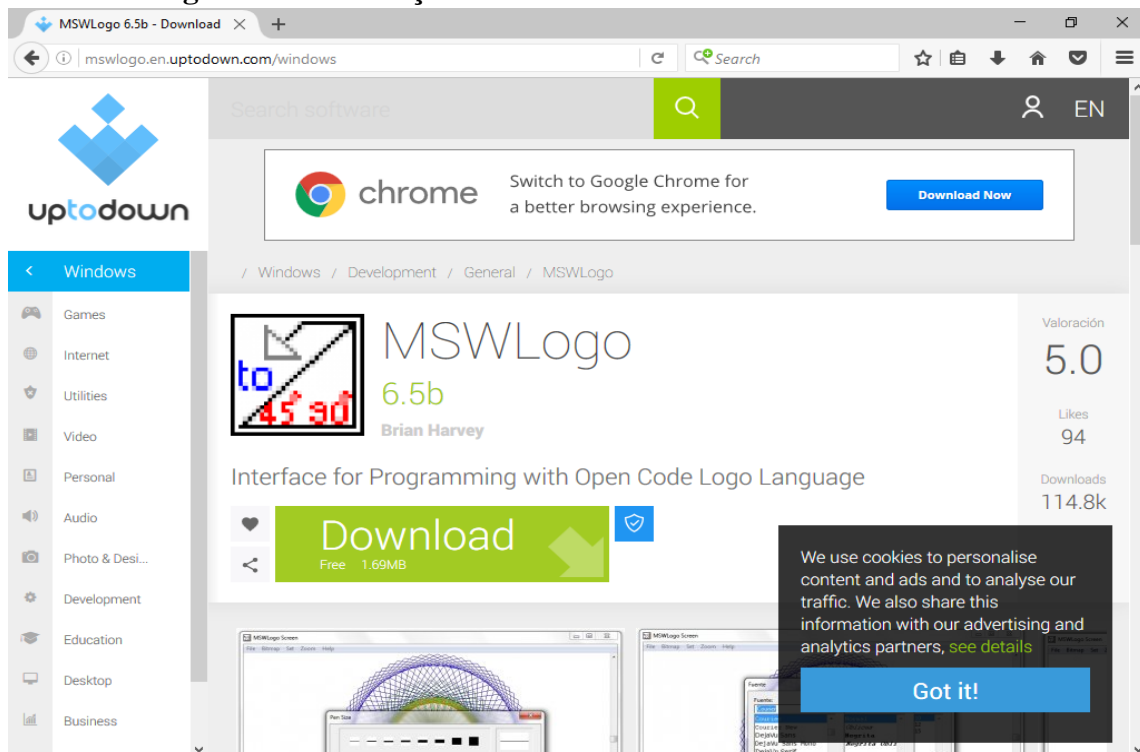
De recordar que o **SuperLogo** os comandos/códigos são da língua portuguesa e o **MSWLogo** da língua inglesa, recomendo **MSWLogo**.

#### 2º Passo

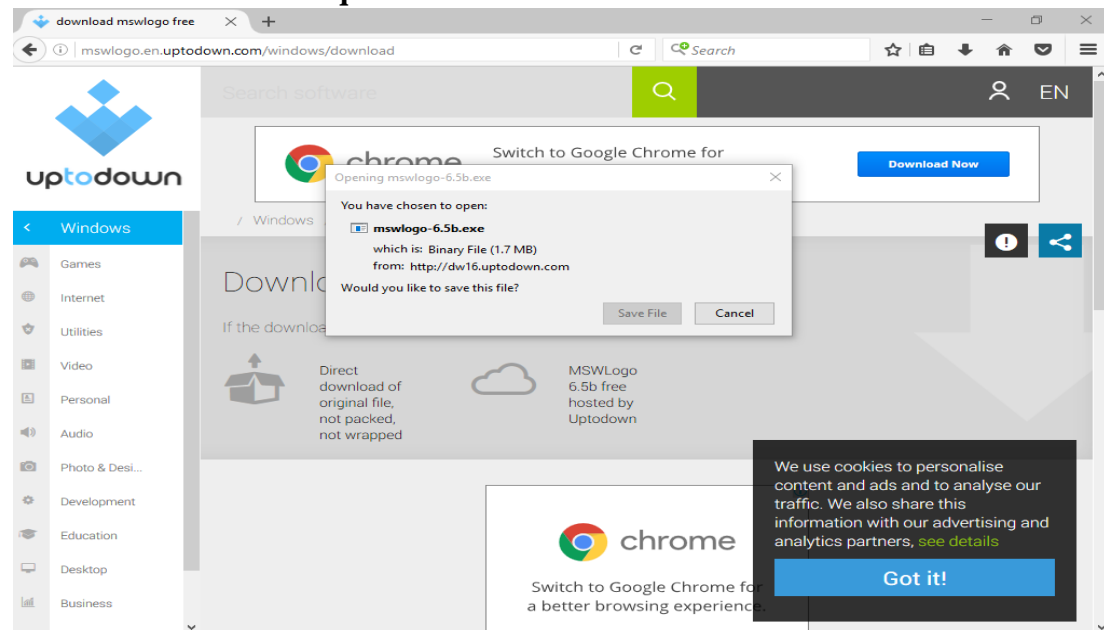
Em vez de pesquisar no motor de busca e pelo nome do compilador pode baixar directamente pelos seguintes endereços:

- ✓ MSWLogo  
<http://mswlogo.en.uptodown.com/windows>  
<http://mswlogo.en.lo4d.com/download>  
<http://qpdownload.com/mswlogo/>

#### 1º Passo – Digitando o endereço



## 2º Passo- Salvando o arquivo.



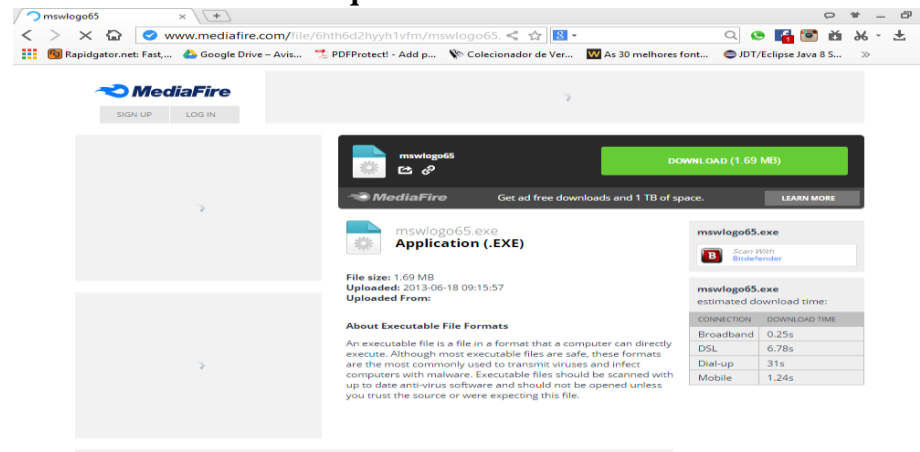
Endereço alternativo:

<http://programandocomsas.blogspot.com/2017/01/mswlogo-compilador.html>

## 1º Passo – Digitando o endereço

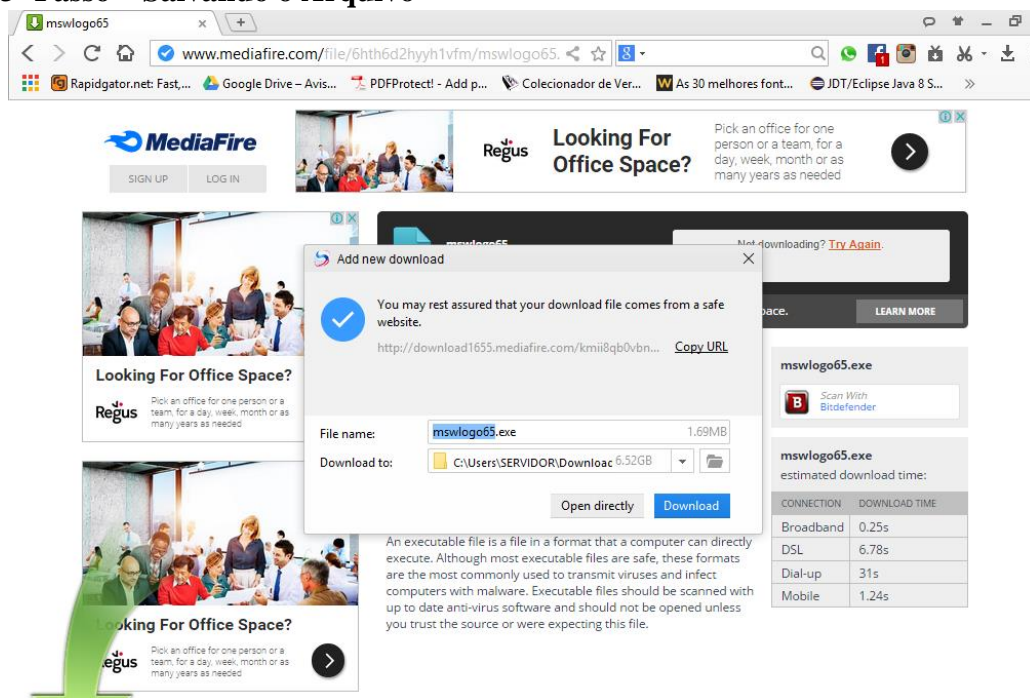


## 2º Passo- Baixando o Arquivo





### 3º Passo – Salvando o Arquivo



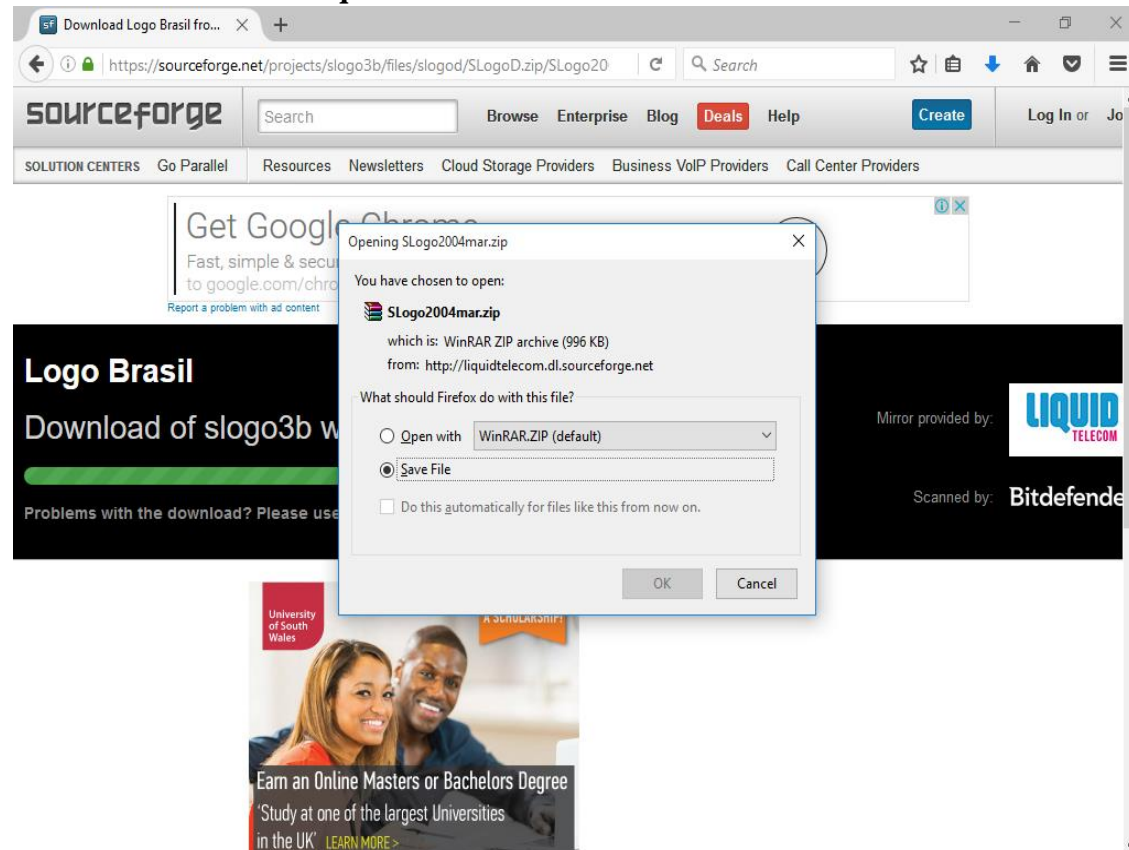
✓ SuperLogo

<http://projetologo.webs.com/slogo.html>

### 1º Passo – Digitando o endereço



## 2º Passo- Salvando o arquivo.



Endereço alternativo:

<http://programandocomsas.blogspot.com/2017/01/superlogo-compilador.html>

## 1º Passo – Digitando o Endereço



## 2º Passo – Baixando o Arquivo

**MediaFire**

SIGN UP LOG IN

**SLogo2004mar** **DOWNLOAD (996.04 KB)**

MediaFire Get ad free downloads and 1 TB of space. LEARN MORE

**SLogo2004mar.zip**  
Archive (.ZIP)

File size: 996.04 KB  
Uploaded: 2017-01-04 03:53:05  
Uploaded From:

**About Compressed Archive Files**

Compressed archives combine multiple files into a single file to make them easier to transport or save on disk space. Archiving software may also provide options for encryption, file spanning, checksums, self-extraction, and self-installation. Zip is the most widely used format, used by the Windows operating system and more recently by OSX as well. RAR is also a very popular and flexible format. Unix uses the tar file format, while Linux uses the tar and gz format.

**SLogo2004mar.zip** can be opened with:

- WinZip for PC
- WinZip for Mac

**SLogo2004mar.zip** estimated download time:

CONNECTION	DOWNLOAD TIME
Broadband	0.14s
DSL	3.89s
Dial-up	18s
Mobile	0.71s

## 3º Passo – Salvando o Arquivo

**MediaFire**

SIGN UP LOG IN

**SLogo2004mar** **DOWNLOAD (996.04 KB)**

MediaFire Get ad free downloads and 1 TB of space. LEARN MORE

**SLogo2004mar.zip**  
Archive (.ZIP)

File size: 996.04 KB  
Uploaded: 2017-01-04 03:53:05  
Uploaded From:

**About Compressed Archive Files**

Compressed archives combine multiple files into a single file to make them easier to transport or save on disk space. Archiving software may also provide options for encryption, file spanning, checksums, self-extraction, and self-installation. Zip is the most widely used format, used by the Windows operating system and more recently by OSX as well. RAR is also a very popular and flexible format. Unix uses the tar file format, while Linux uses the tar and gz format.

**SLogo2004mar.zip** can be opened with:

- WinZip for PC
- WinZip for Mac

**SLogo2004mar.zip** estimated download time:

CONNECTION	DOWNLOAD TIME
Broadband	0.14s
DSL	3.89s
Dial-up	18s
Mobile	0.71s

**Download** dialog box:

You may rest assured that your download file comes from a safe website.  
<http://download857.mediafire.com/c7dd0084j14g/t...> **Copy URL**

File name: SLogo2004mar.zip 996.04KB

Download to: C:\Users\SERVIDOR\Downloads 5.52GB

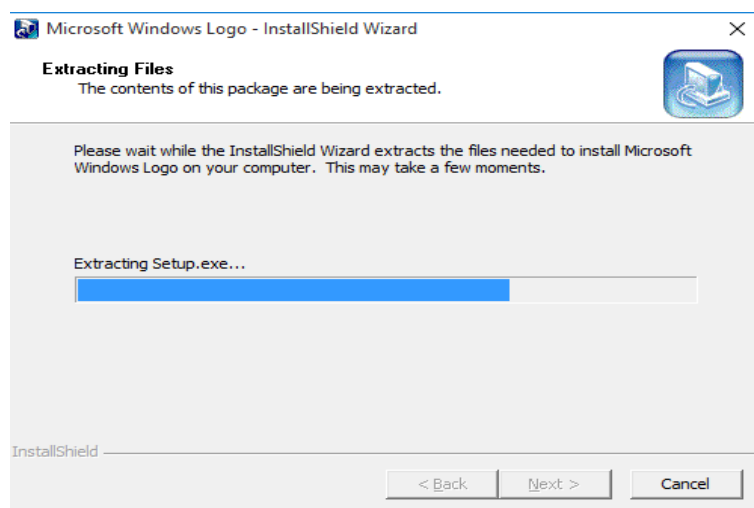
**Open directly** **Download**

## Instalação dos Compilador

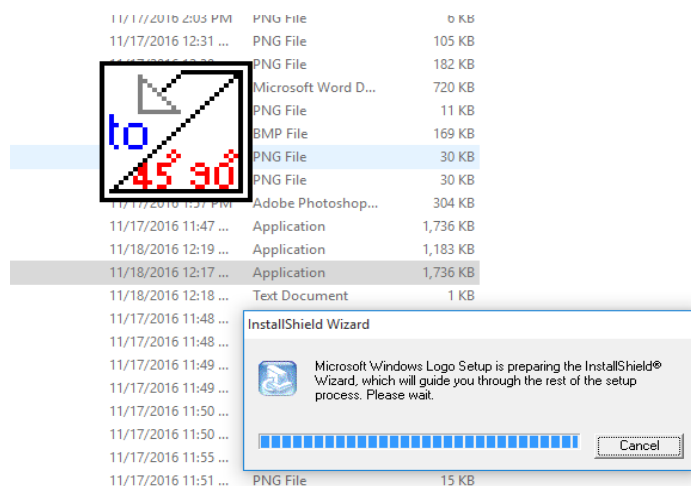
## Instalação de MSWLogo

Para fazer a instalação de MSWLogo é só seguir os seguintes passos:

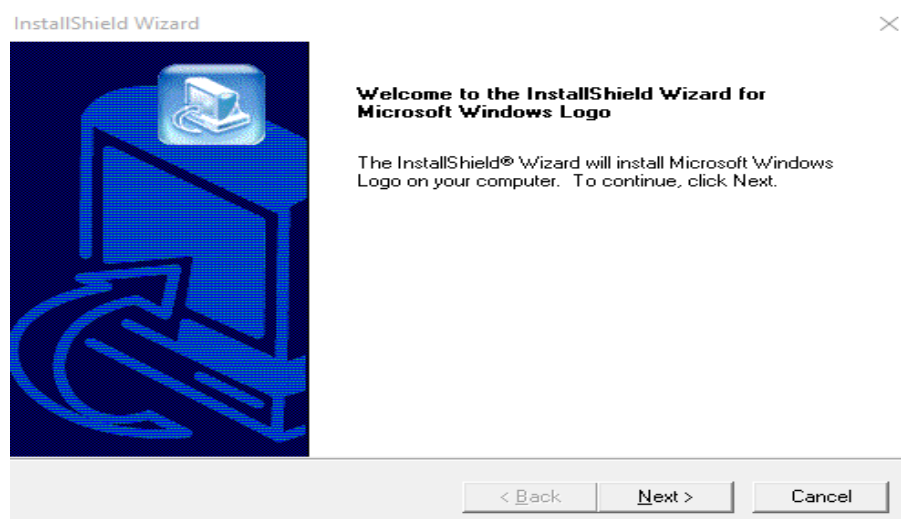
## 1º Passo – Abra o arquivo baixado



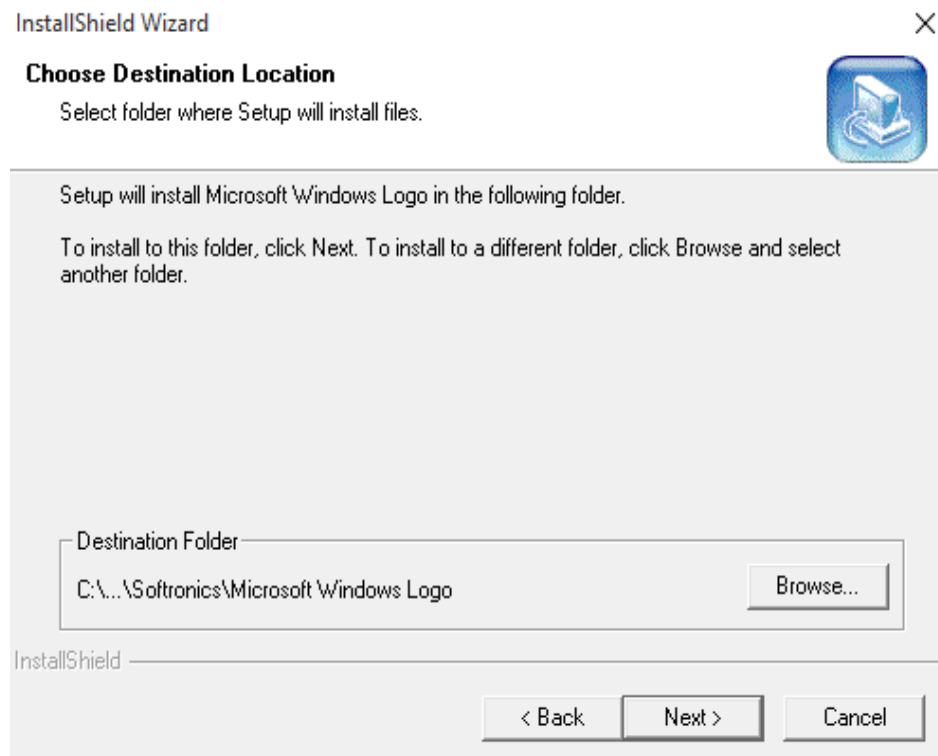
## 2º Passo - Espere o ficheiro descompactar



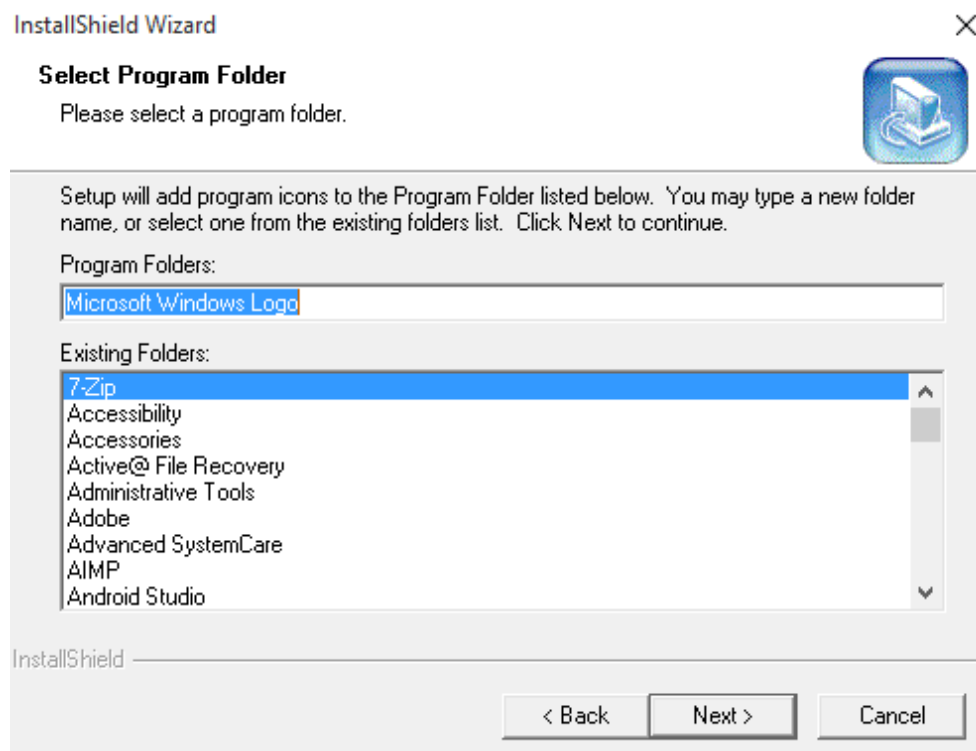
## 3º Passo – Pressione “Next”



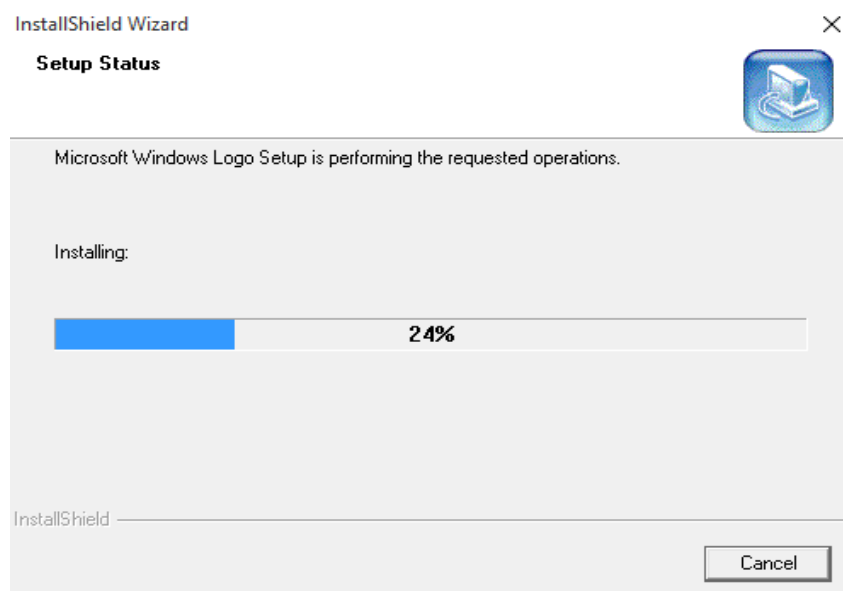
#### 4º Passo – Escolha o local para guardar, recomendo o padrão. Depois pressione “Next”



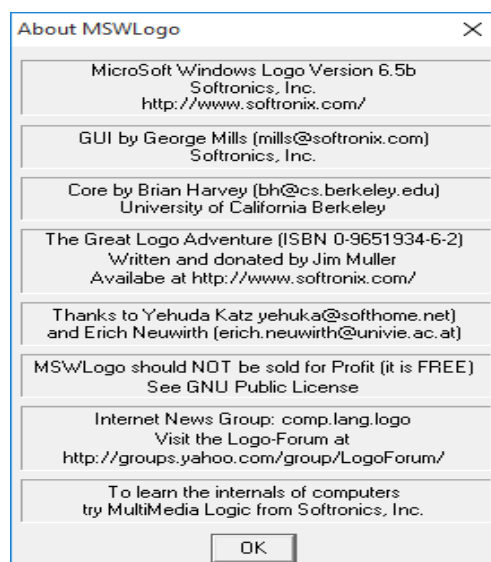
#### 5º Passo – Pressione “Next”



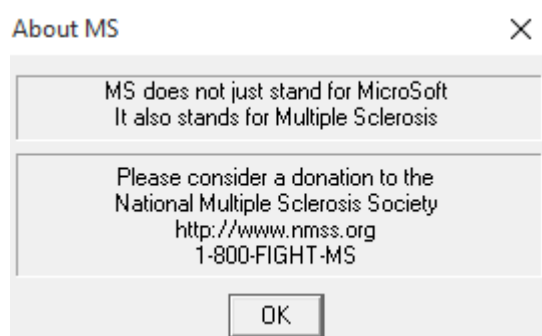
## 6º Passo – Aguarde o processamento



## 7º Passo – Pressione “OK”



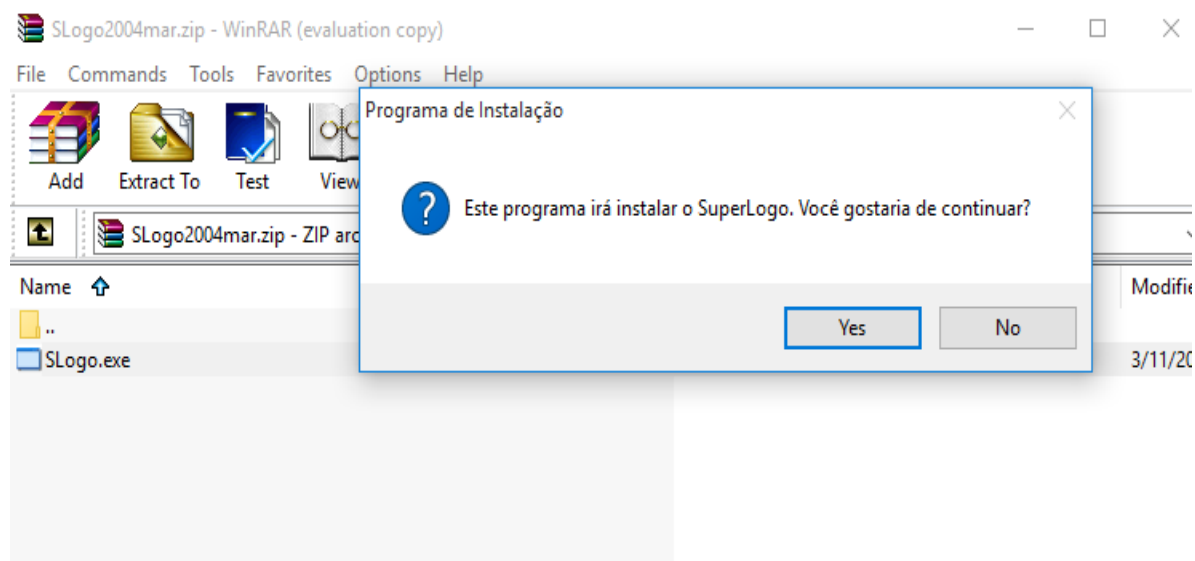
## 8º Passo – Pressione “OK”



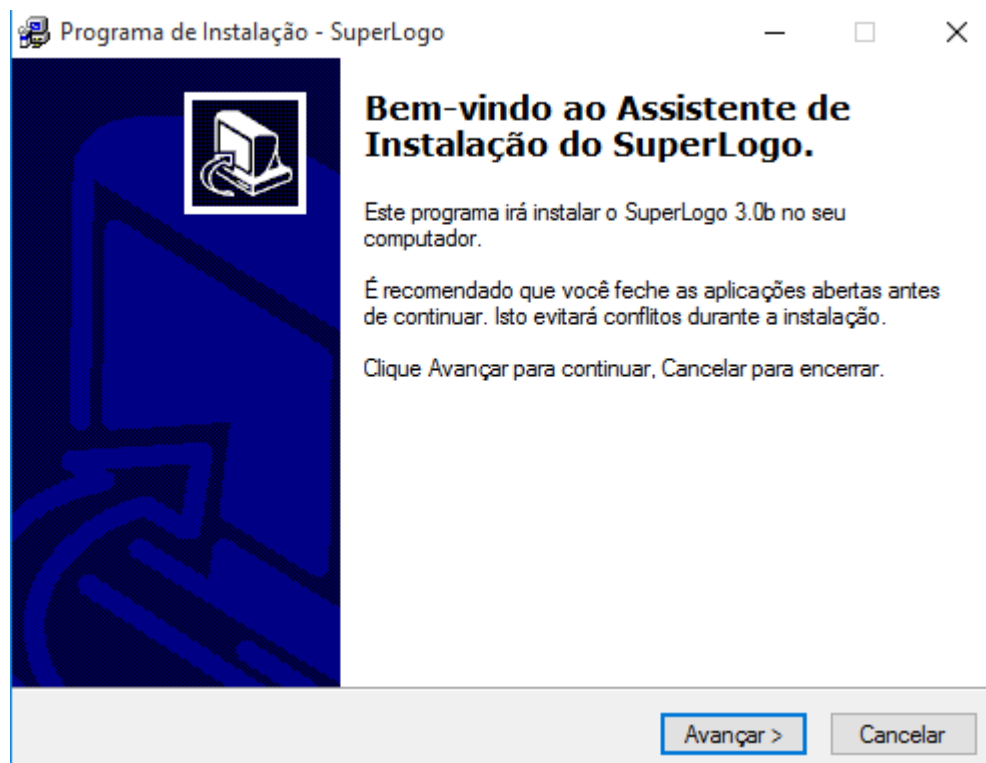
## Instalação de SuperLogo

Para fazer a instalação de SuperLogo é só seguir os seguintes passos:

### 1º Passo – Abra o arquivo baixado

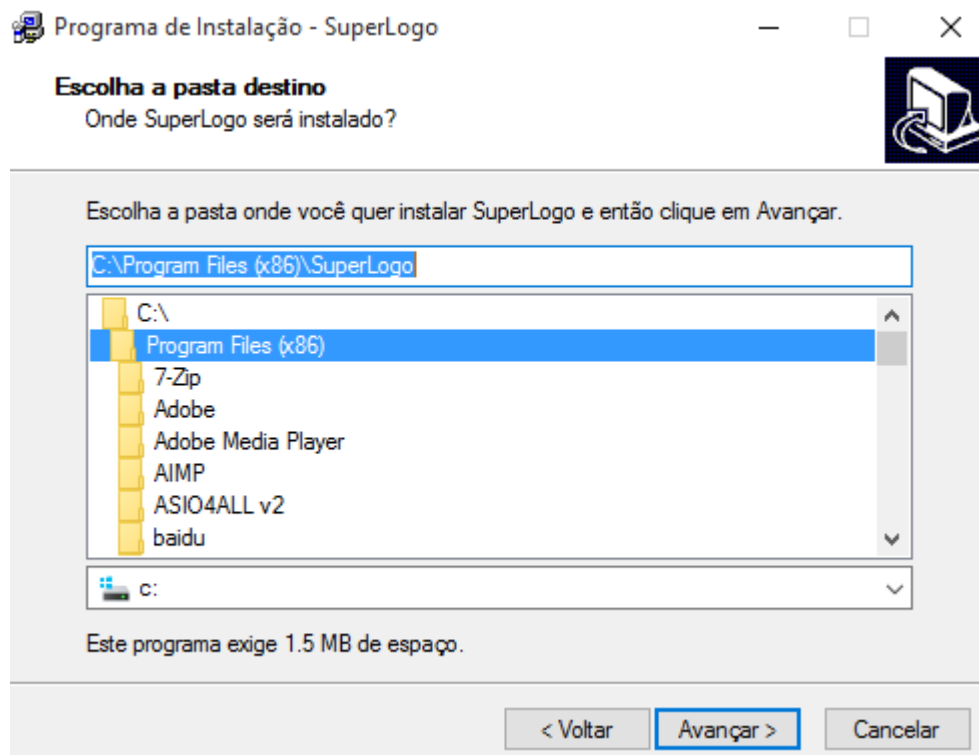


### 2º Passo – Pressione em “Avançar”

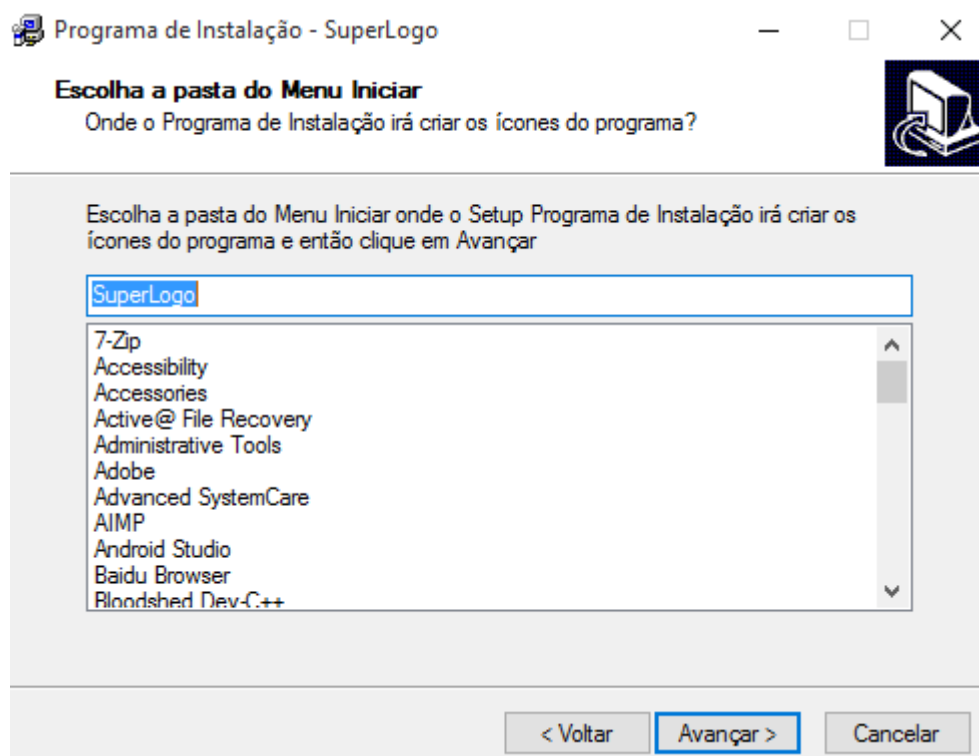




### 3º Passo – Escolha o local para guardar, recomendando o padrão. Depois pressione “Avançar”

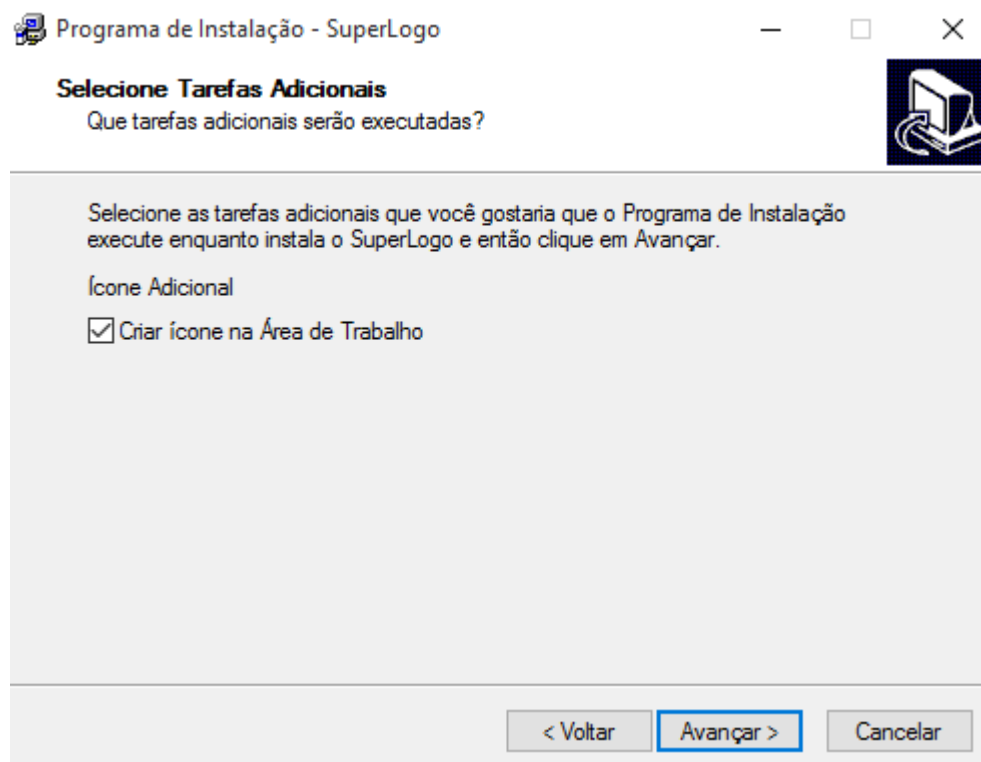


### 4º Passo – Pressione em “Avançar”

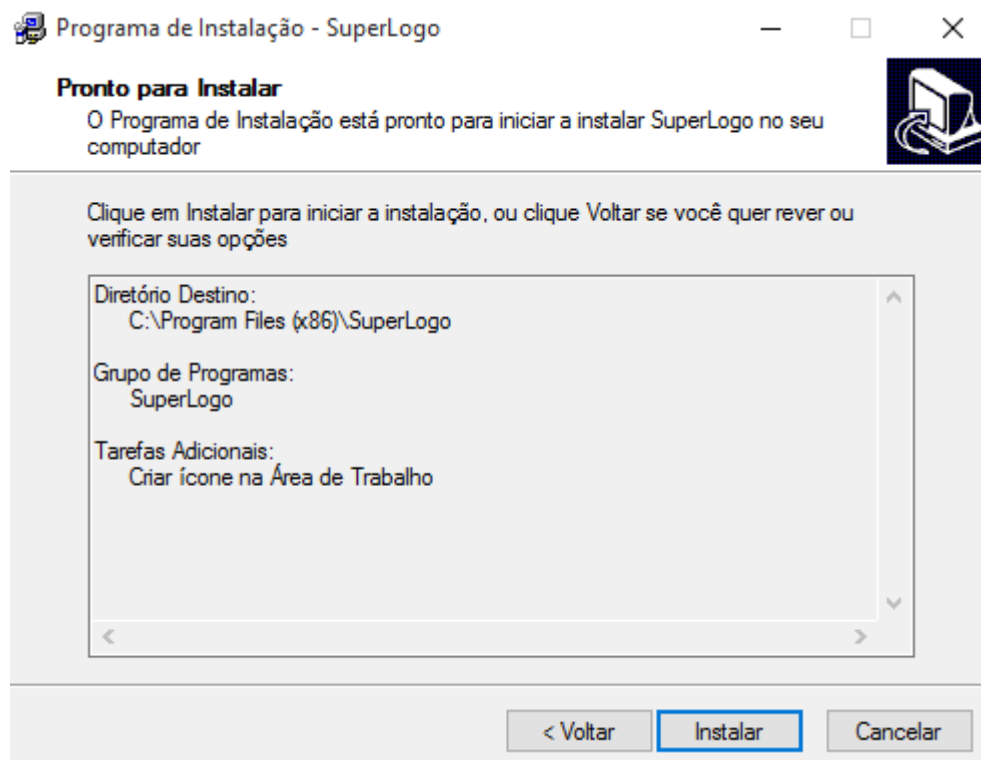




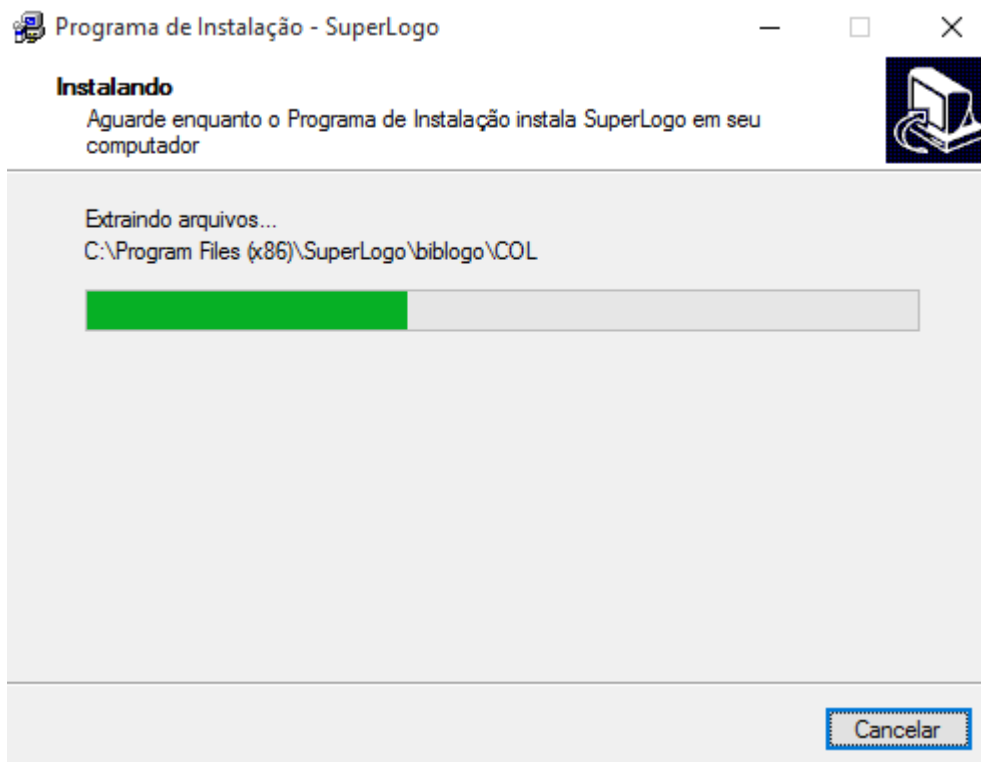
## 5º Passo - Pressione em “Avançar”



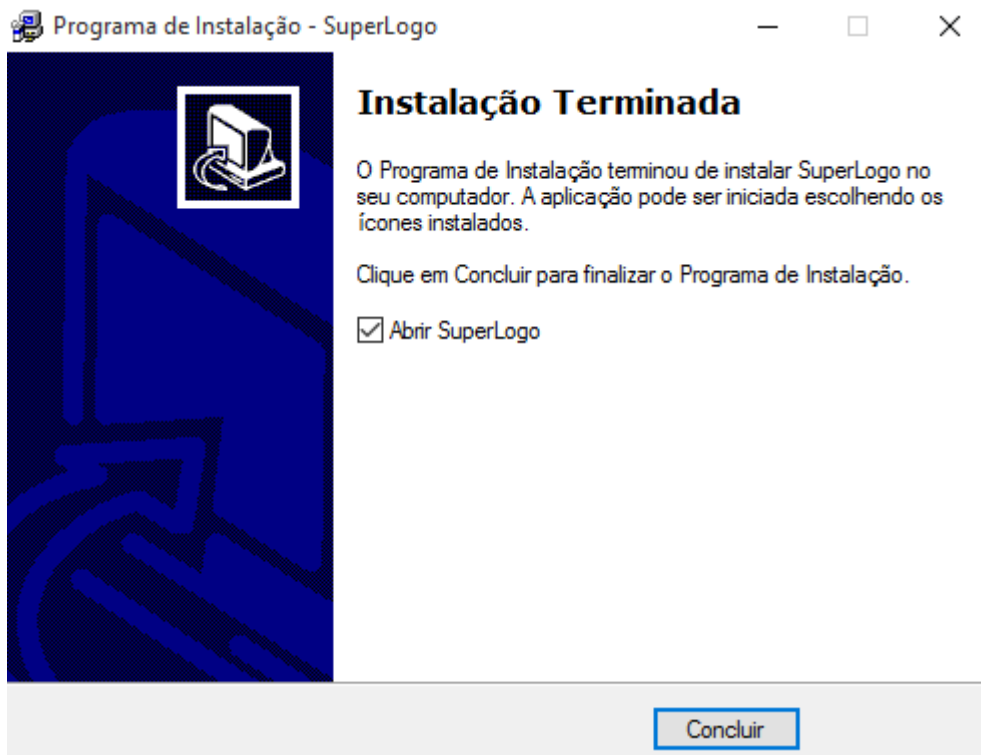
## 6º Passo - Pressione em “Instalar”



## 7º Passo – Aguarde o Processamento



## 8º Passo - Pressione em “Concluir” com a opção “Abrir SuperLogo” selecionado



## 9º Passo – Pressione em “Não”

SuperLogo 3.0



O mneumônico "ap" do comando aprenda não está disponível nesta versão.

Caso você queira carregar um arquivo que contenha esse mneumônico, abra-o antes em um editor de texto.

Substitua as ocorrências de "ap" por aprenda e depois carregue-o no SuperLogo 3.0.

Deseja que esta mensagem continue a ser exibida todas vez que o SuperLogo 3.0 for aberto ?

Yes

No

## Capítulo III

### Comandos Básicos

Os comandos primitivos em linguagem **logo** são:

Primeiramente mostraremos a tabela dos comandos à Português e depois só mostraremos os comandos a Inglês por ser o padrão da programação.

Comando	Significado	Exemplo
PF <Número>	Para Frente	PF 22
PD <Número>	Para Direita	PD 80
PE <Número>	Para Esquerda	PE 120
PT <Número>	Para Trás	PT 12
REPITA <Número> [Instrução]	Repita	REPITA 4 [PF 40 PD 90]

Comando	Significado	Exemplo
FD <Número>	Para Frente	FD 22
RT <Número>	Para Direita	RT 80
LT <Número>	Para Esquerda	LT 120
BK <Número>	Para Trás	BK 12
REPEAT <Número> [Instrução]	Repita	REPEAT 4 [FD 40 RT 90]

### Estrutura de um Programa em Logo

Código na língua Portuguesa:

**aprenda <nome do programa>**

**instruções**

**fim**

Exemplo prático:

**aprenda quadrado**

**repita 4 [pf 120 pd 90]**

**fim**

Código na língua Inglesa:

**to <nome do programa>**

**instruções**

**end**

Exemplo prático:

**to quadrado**

**repeat 4 [fd 120 rt 90]**

**end**

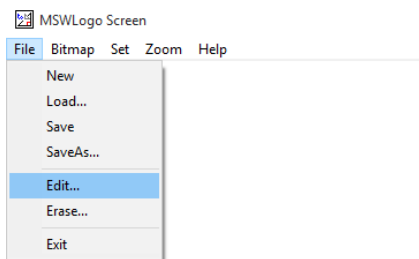
## Visualização do Programa em Logo

Para visualizar um programa em logo basta escrever o seguinte comando:

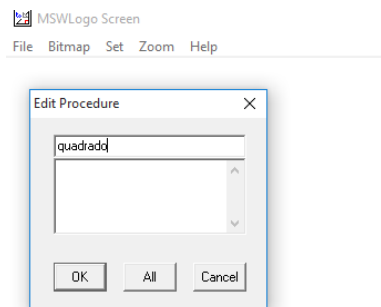
**show <nome do programa>**

## Demonstração de criação e visualização de um Programa em Logo

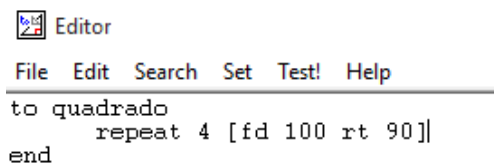
### 1º Passo – *File<Edit*



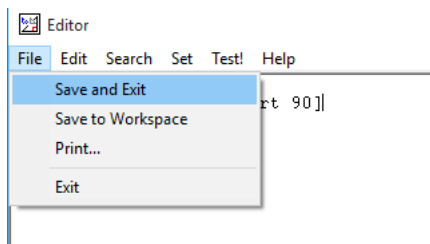
### 2º Passo – *Escreva o nome<OK*



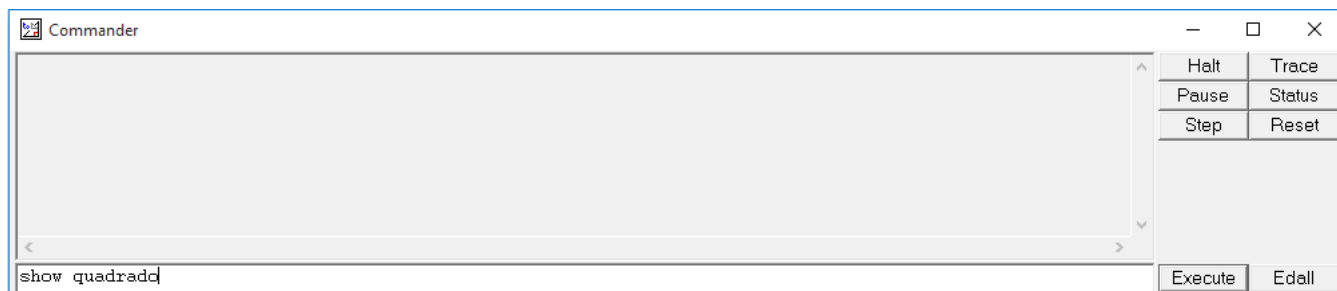
### 3º Passo – *Escreva o código/comando*



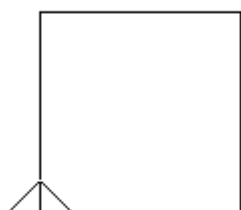
#### 4º Passo – *File<Save and Exit*



#### 5º Passo – *Escreva o comando de visualização<Execute*



#### 6º Passo – *Resultado*



### Outros comandos

Depois da tabela com a lista dos comandos primitivos, agora mostraremos a tabela com outros comandos usados, mas estes comandos estarão na língua inglesa e o significado na língua portuguesa.

Comando	Significado	Exemplo
PU [Instrução]	Use Nada	PU FD 12
PD [Instrução]	Use Lápis	PD RT 45
ARC <Ângulo> <Raio>	Arco	ARC 60 4
PE [Instrução]	Use Borracha	PE FD 30

PN [Instrução]	Voltar ao normal	PN RT 30
SetPenSize [número]	Tamanho do risco	SetPenSize [12 10]
SetPC [r g b]	Usar Cor do Lápis	SetPC [2 5 5] ou setpc 8
setfloodcolor [r g b]	Usar Cor no Interior	Setfloodcolor [1 2 5]
ST	Mostrar a tartaruga	ST LT 20
HT	Ocultar a tartaruga	FD 20 RT 45 HT
CS	Limpar a Tela	
WAIT <Número>	Espera alguns segundos	WAIT 60
RANDOM <Número>	Retorna um valor aleatório com base no número dado (o número esta dentro do intervalo)	RANDOM 12
FOR [i start stop step] [Instrução]	Ciclo de Repetição	FOR [i 0 12 2] [SHOW :i]
PRINT	Mostrar na tela	PRINT "LOGO
OUTPUT (OP)	Mostrar na tela	OP [4]
WHILE [Condição] [Instrução]	Ciclo de Repetição	WHILE [not empty :nome] [output :nome]
;	Comentário	;chamada de ciclo for
EMPTY	Comando usado para dizer VÁZIO	IF EMPTY :NOME [OUTPUT 0]
<b>Mais Comando</b>		
BUTFIRST (BF)	PX	WINDOWS
BUTLAST (BL)	SETBG	WRAP
LAMBDA ( $\lambda$ )	SETPN	HEADING
!	POTS	POS
SQRT	POPS	XCOR
TRUE	PO	YCOR
FALSE	ERASE (ER)	EDSH
FIRST	ERPS	TELL
LAST	PR	SETSH
BG	SETH	PUTSH
PC	SETPOS	SETC
SETSP	WHEN	GETSH
IF	STOP	KEYP
RC	SPEED	=
PPT	HOME	BYE
TO	THING	CLEANTEXT
EDITFILE	READWORD	REPCOUNT
SENTENCE	LABEL	SETH

Para o comando **FOR** deve-se destacar que:

- ✓ **i**, é uma variável, contém o valor actual do contador.
- ✓ **start**, indica por onde começar,
- ✓ **stop**, indica onde parar, e
- ✓ **step**, indica como a etapa deve ser.

### Exercícios

1. Faça um programa que imprime na tela o seu primeiro nome.
2. Faça um programa que imprime um quadrado, lado igual à 120.
3. Faça um programa que imprime um retângulo, largura igual à 60 e comprimento igual à 120.
4. Faça um programa que imprime um círculo, cujo raio é 90.
5. Faça um programa que imprime um triângulo equilátero, cujo lado é igual à 140.
6. Faça um programa que imprime um triângulo cujo valor de raio é digitado pelo usuário.
7. Faça um programa que imprime uma figura com números de lados dado pelo usuário.
8. Faça um programa que imprime a bandeira de Moçambique.
9. Faça um programa que mostra uma palhota.
10. Faça um programa que mostra uma casa de alvenaria.
11. Crie um programa que imprime uma escada com 6 degrau.
12. Crie um programa que imprime uma aranha.
13. Faça um programa que imprime um homem.
14. Faça um programa que imprime uma árvore.
15. Faça um programa que imprime um cisne.
16. Faça um programa que imprime na tela o resultado de qualquer função linear.
17. Faça um programa que imprime na tela o resultado de qualquer função quadrática.
18. Sem usar qualquer compilador diga qual será o output das seguintes expressões:
  - Show butfirst first [[a b][c d][e f]]
  - Show first butfirst [[a b][c d][e f]]
  - Show first butfirst [3 7 5]
  - Show last [a [12 c][b f] 123 last]
  - Dado o seguinte programa:

```
to pense :lista
  ifelse empty? :lista [output 0]
  output (first :lista) + (pense butfirst :lista)
end
```
- Diga qual será o output:
  - ✓ Show pense [12 7 8]
19. Crie um programa que imprime na tela a soma de dígitos de um dado número.
20. Crie um programa que conta elementos numa lista.
21. Faça um programa que calcula a média de duas notas digitadas pelo utilizador.
22. Faça um programa que mostra o valor absoluto do número dado pelo usuário.
23. Crie um programa que faz o cálculo de Fibonacci com base no número dado.
24. Crie um programa que imprime o dobro do número dado pelo usuário.
25. Faça um programa que diz se o número introduzido é PAR ou IMPAR.
26. Faça um programa que calcula o m.m.c.
27. Faça um programa que calcula o m.d.c.



28. Crie um programa que compara dois números e imprime o maior deles.
29. Crie um programa que calcula a potência de um número  $a^b$ .
30. Sem usar um compilador diga qual será o output:
- $(+ 3 (* 7 6) 5 (/ 4 2))$
  - $(+ 42 12 9)$
  - $(* 0 (/ 10 2) 7 (+ (- (* 13 2) 9) 87))$
31. Crie um jogo qualquer.
32. Converta a função  $f(x) = x^3 + 4$  para a notação Lambda e encontre o resultado de  $f(2)$ .
33. Sem usar um compilador diga qual será o output de:
- $(\text{lambdaX}.\text{lambdaY} . + X ((\text{lambdaX} . - X 3) Y)) 9 7$ .
34. Escreva uma definição da função crescente :: [Int] ! Bool que verifica se uma lista de inteiros está por ordem crescente. Exemplos: crescente [43; 12; 60] = False; crescente [2; 4; 6; 8] = True. Deve considerar que a lista vazia e as listas com um só elemento estão por ordem crescente.

## Referência Bibliográfica

<https://pt.wikipedia.org/wiki/Logo>, acessado no dia 17 de Novembro de 2016, as 15h28min;

SINGO, F., Concepts of Programming Languages, Programação Funcional, Universidade Pedagógica, Departamento de Informática, 2016;

BASILIO, C., Languages de Programação, Programação Funcional, Universidade Federal Fluminense;

SINGO, F., Concepts of Programming Languages, Programming with MSW LOGO, Universidade Pedagógica, Departamento de Informática, 2016;

Atari Logo, Introduction to Programming, Throught Turtle Graphics, Canada, 1983;

DUCHARME, Bob, Logo For Kids: Na Introduction, 2002;



### Sualé Abdul Sualé

Moçambicano, natural da Cidade de Maputo. Fez o ensino técnico profissional na Escola Comunitária Artes e Ofícios, em Moçambique, província da Zambézia, cidade de Quelimane, onde fez o curso básico de Informática por três anos de duração, onde este curso concluiu em 2012. Em 2014, terminou o ensino pré-universitário na Escola Secundária 25 de Setembro, na província da Zambézia, cidade de Quelimane. Actualmente é estudante universitário, onde está a fazer Licenciatura em Informática na Universidade Pedagógica - Moçambique, província de Maputo, cidade de Maputo. Inovador, curioso, amante da música e do desporto (Hóquei em

Patins, em particular), ele aprende e ensina a cada dia, não esquecendo que sempre que encontra um impasse lembresse da sua frase de motivação “*Não existe problema sem solução*”, sendo essa a chave do seu sucesso.

## Anexos

Ordem alfabética dos comandos mais usados em Logo.

<b>A</b> ARC ARCL ARCL1 ARCLEFT ARCR ARCR1 ARCRIGHT <b>B</b> BACK BG BK BOXR BREAK bug <b>C</b> CAPS LOWR CATALOG CB CIRCLE CIRCLEL CIRCLES color table command CROSS  CS CT CTRL → CTRL ← CTRL DELETE BACK S CTRL A CTRL E CTRL F CTRL ↓ CTRL ↑ CTRL S CTRL T CTRL Y cursor <b>D</b> debug DELETE BACK S DIAMOND DIAMONDS <b>E</b> ED EDIT EDISH END ER ERASE ERF ERPS ESC <b>F</b> FD file FLAG FLAGBACK FLAGS FLOWER FORWARD FS	<b>G</b> GAME GETSH GREET GREET1 <b>H</b> HEADING hexagon HOME HOUSE HT <b>I</b> IF input <b>K</b> KEYP <b>L</b> LEFT LEFTLEG LEFTSIDE LISTEN LOAD LT <b>M</b> MAKE MAN MANYFLAGS <b>O</b> OVER operation <b>P</b> PC PD PE PENDOWN PENUP PLAY  PN PO POLY POLY1 POPS POS position POTS PR primitive PRINT procedure prompt PU PUTSH PX <b>R</b> RANDOM RC recursive REPEAT RETURN RIGHT RIGHTLEG RIGHTSIDE RT RULES <b>S</b> SAVE SETBG SETC SETGAME SETH SETPC	SETPN SETPOS SETSH SETSP SETTREE SETUP SHIFT DELETE BACK SPACE BAR SPEED SPI SPIDER SPINFLAG SPINSTAR SQUARE SQUARES SQUARESTAR ST SS state STOP SWAN SWIRL SYSTEM RESET <b>T</b> TELL TENT title line TO TOUCHING TREE TREES TRI TRIANGLE TRIANGLER TRIANGLES TRISTAR TS turtle  <b>V</b> variable <b>W</b> WELL WHEN WINDOW workspace WRAP <b>X</b> XCOR <b>Y</b> YCOR [ ]
---	---	---

Tabela com Cores mais usadas.

<b>Name</b>	<b>Index</b>	<b>RGB values</b>	<b>Name</b>	<b>Index</b>	<b>RGB values</b>
Black	0	[ 0 0 0 ]	Brown	8	[155 96 59]
Blue	1	[ 0 0 255]	Light brown	9	[197 136 18]
Green	2	[ 0 255 0 ]	Mid-green	10	[100 162 64]
Cyan	3	[ 0 255 255]	Blue-green	11	[120 187 187]
Red	4	[255 0 0 ]	Salmon	12	[255 149 119]
Magenta	5	[255 0 255]	Blue-ish	13	[144 113 208]
Yellow	6	[255 255 0 ]	Orange	14	[255 163 0 ]
White	7	[255 255 255]	Silver	15	[183 183 183]